

Theoretical analysis of library screening using a N-dimensional pooling strategy

Emmanuel Barillot^{1,2}, Bruno Lacroix¹ and Daniel Cohen^{1,2}

¹Centre d'Etude du Polymorphisme Humain (CEPH), 27 rue Juliette Dodu, F-75010 Paris and

²GENETHON, 13 place de Rungis, F-75013 Paris, France

Received August 6, 1991; Revised and Accepted October 15, 1991

ABSTRACT

A solution to the problem of library screening is analysed. We examine how to retrieve those clones that are positive for a single copy landmark from a whole library while performing only a minimum number of laboratory tests: the clones are arranged on a matrix (i.e. in 2 dimensions) and pooled according to the rows and columns. A fingerprint is determined for each pool and an analysis allows selection of a list containing all the positive clones, plus a few false positives. These false positives are eliminated by using another (or several other) matrix which has to be reconfigured in a way as different as possible from the previous one. We examine the use of cubes (3 dimensions) or hypercubes of any dimension instead of matrices and analyse how to reconfigure them in order to eliminate the false positives as efficiently as possible. The advantage of the method proposed is the low number of tests required and the low number of pools that require to be prepared [only 258 pools and 282 tests (258 + 24 verifications) are needed to screen the 72,000 clones of the CEPH YAC library (1) with a sequence-tagged site]. Furthermore, this method allows easy and systematic screenings and can be applied to a large physical mapping project, which will lead to an interesting map with a low, precisely known, rate of error: when fingerprinting a 150 Mb chromosome with the CEPH YAC library and 1750 sequence-tagged sites, 903,000 tests would be necessary to obtain about 20 contigs of an average length of 6.7 Mb, while only about one false positive would be expected in the resultant map. Finally, STSs can be ordered by dividing a clone library into sublibraries (corresponding to groups of microplates for example) and testing each STS on pooled clones from each sublibrary. This allows to dedicate to each STSs a fingerprint that consists in the list of the positive pools. In many cases these fingerprints will be enough to order the STSs. Indeed if large YACs (>1 Mb) can be obtained, the combined screening of DNA families and YAC DNA pools would allow an integrated construction of both genetic and physical maps of the human genome, that will also reduce the optimal number of meioses needed for a 1 centimorgan linkage map.

INTRODUCTION

Screening of a clone library is a preliminary indispensable to numerous genetic studies and is usually performed with a probe or a sequence-tagged site (STS: see 2). It requires a large number of manipulations if the test is done on each clone individually and several strategies have been proposed in order to minimize work (3 to 6): the first idea is to divide the library into several subgroups (3 & 4) and to perform the tests on each subgroup; the next problem is to retrieve the positive clone(s) among the positive subgroup(s). This approach noticeably reduces the number of tests, but when a large number of screenings are performed, the number of different pools to be assembled is very high. The pooling strategy (5 & 6) has the advantage of requiring only a small number of different pools, tests on which allow retrieval of the positive clones. However, this strategy has not been analysed theoretically for optimization, thus the purpose of the present study. First, the theoretical principles of the method are presented. Second, practical considerations are exposed. Then the results are analysed and the method is compared with a modification of a previous pooling strategy (3, 4). Finally, we propose to apply this method in order to construct the physical map of a large genome and we expose the help that this method could provide for genetic mapping.

In synthesis, the subject of this article can be considered as the experimental design of the following problem: an ensemble contains a large number of elements, of which only a few are positive; a test performed on a subset of element gives a binary answer that is positive if, and only if, there is at least one positive element in the subset, otherwise negative. The purpose is to retrieve all the positive elements.

For all calculations, we assumed the cloning step to be perfectly random (this could not be strictly verified because of any possible cloning bias, but should serve as a good approximation); it directly follows that the number of positive elements has a Poisson distribution whose mean equals the redundancy of the library. All mathematical considerations are deferred until the last section.

THE POOLING STRATEGY: THEORETICAL PRINCIPLES

The 2-dimensional pooling strategy

This method of clone pooling consists of a multiplex approach (analogous to the method described by De Jong (5) and Evans (6)): if we imagine that all the clones from the library are arranged in a 2-dimensional matrix, we can then pool them according to the rows and the columns of the matrix (each row and each

column gives a pool, thus two copies of the library are needed). Next we can perform a test on each pool (there are now $2N^{1/2}$ tests instead of N , a noticeable reduction). A positive clone will render its row and its column positive, so all the positive clones will be located at the intersection of a positive row and a positive column. But the reverse is not true: an intersection of a row containing a positive clone with a column containing a different positive clone does not necessarily give a positive clone. In fact, if P clones are positive, then P rows and P columns are generally positive (but this is not always true, since two positive clones can be in the same row or column), and P^2 intersections are potentially positive (i.e. they are candidates), but only P intersections are effectively positive and it is not possible to retrieve them directly. To discriminate between true and false positives, one or several other matrices are needed, since the true positives are inevitably present as candidates in every matrix. The new matrices have to be configured in a way as different as possible from the previous ones, so that a given illegitimate candidate from a given matrix is not likely to be retrieved as an illegitimate candidate in another matrix, and thus can be detected. Finding the optimal configuration which will minimize the number of illegitimate candidates is in itself a difficult problem which will be dealt with in the third section. Obviously, the higher the number of positive clones, the higher the number of matrices needed to eliminate all the illegitimate candidates. Typically, this number varies from 2 to 8, depending on the redundancy (as the average number of positive clones per STS equals the redundancy), the number of clones, and the tolerated rate of false positives.

The N-dimensional pooling strategy

In order to further reduce the number of pools, it is possible to arrange clones in a 3-dimensional space (a cube) and pool them according to the $3N^{1/3}$ possible planes of the cube. Candidates are still located at the intersection of the positive planes, thus P^3 candidates per cube are expected and, as in the case of two-dimensional arrays, several cubes are needed to eliminate the false candidates. The generalization at any D -dimensional space is immediate: the number of pools is the number of hyperplanes, i.e. $DN^{1/D}$ and the number of candidates is P^D . The number of pools decreases with the dimension (while $D < \ln(N)$) but the number of candidates, and therefore the number of false positives among the candidates, increases and so the number of configurations (i.e. matrices, cubes or hypercubes) required to eliminate them also increases. The balance of these two effects gives the optimal pooling dimension, which is dependent on the total number of clones fingerprinted and the redundancy of the library (Fig. 1).

Vocabulary definitions

It is useful to define some essential words: the pooling dimension D is the dimension of the space in which clones are conceptually arranged before pooling ($D = 2$ for a matrix, 3 for a cube, etc); a configuration is, depending on the dimension, a matrix, cube or hypercube which has been used to arrange the clones before pooling; A scheme (5) is a set of pools which contains each clone once, and only once (in 2 dimensions, the rows of the matrix define a scheme, as the columns), and the side length of a scheme is its number of pools (in 2 dimensions, the side length is the matrix side length; in 3 dimensions, it is the cube side length; etc.). Therefore a scheme represents one copy of the library while a configuration corresponds to D schemes and D copies of the

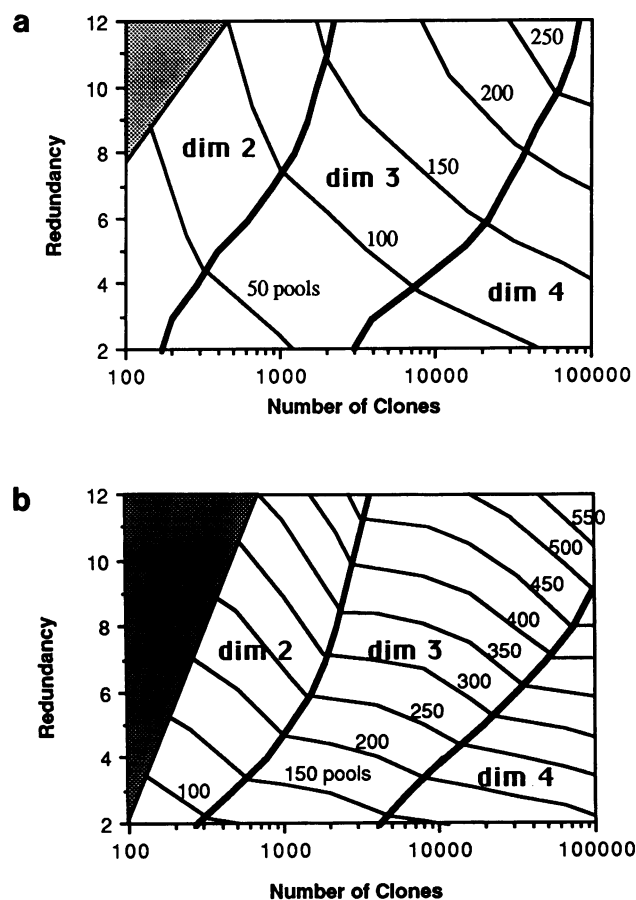


Fig. 1. Optimal pooling dimension and number of pools required for screening as a function of the redundancy of the library and the number of clones analysed. The horizontal axis is graduated in logarithmic units. The grey area defines a region where the pooling strategy is not interesting; the thick lines separates the regions where the optimal pooling strategy is, respectively, 2, 3 and 4; the thin lines give the required number of pools. The optimal pooling dimension and the number of pools depend not only on the redundancy and the number of clones, but also on the desired rate of false positives. The figure 1.a tolerates as many false positives as true positives: it is useful for general library screening, when it is possible to verify individually each selected positive clone. The figure 1.b is for 0.01 false positives per screening, a good rate for physical mapping. For example, for the CEPH library which contains 72,000 YACs and is 10-fold representative, the optimal pooling dimension is 3 (but the dimension 4 gives almost equivalent results). Clones will then be pooled according to 3-dimensional cubes with edge length 43 clones. If the tolerated rate of false positives is 10, about 260 pools will be needed; if the rate is 0.01, about 510 pools are required. Proof of all the calculations are available upon request.

library. The side length equals $N^{1/D}$, where N is the number of clones and a pool contains $N/N^{1/D} = N^{(D-1)/D}$ clones.

False positives

The drawback of the pooling strategy compared to non-pooling approaches is that it introduces the possibility of false positives. The use of additional configurations allows their rate to be reduced, but there is still a chance of a given clone being present as an illegitimate candidate in every configuration and thus of it being wrongly selected as positive. This aspect is not negligible, nor is it dramatic since: 1) the expected number of false positives is easily computable (see sections 4 and 5); 2) in some cases, it is certain that there are no false positives: Indeed the number of true positives is inevitably greater than the number of positives

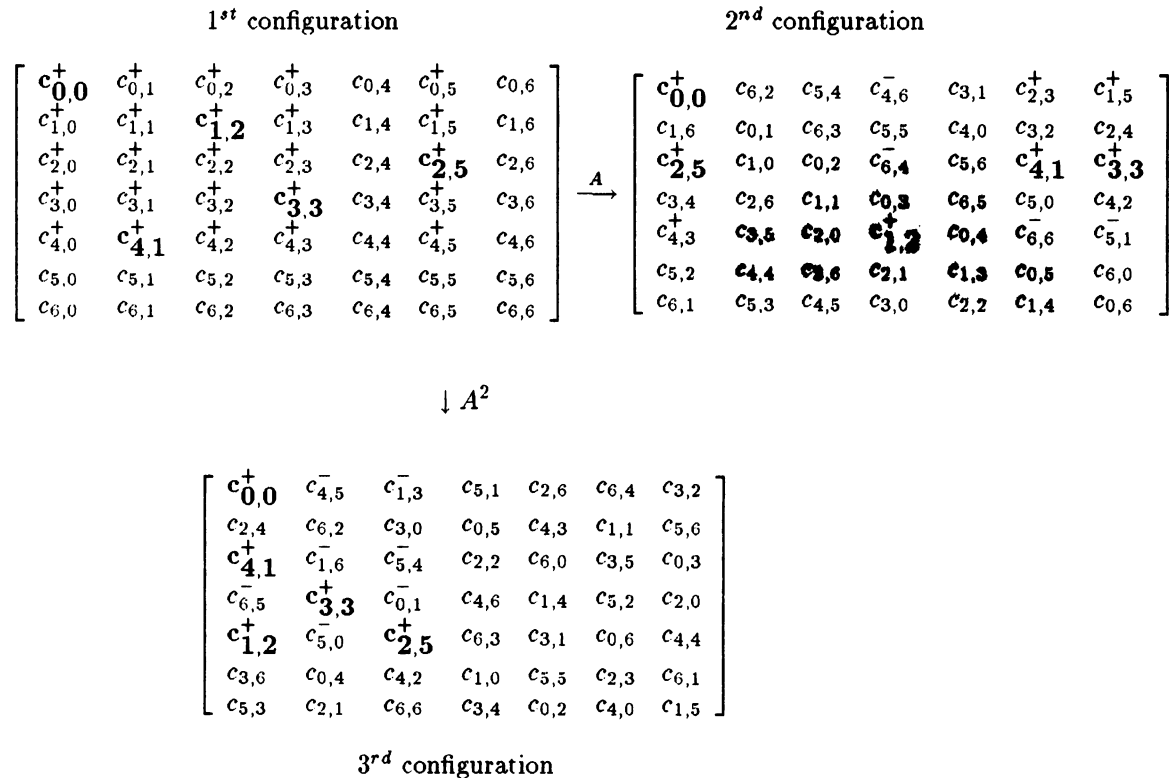


Fig. 2. Example of reconfiguration: a 49 clones library is arranged in a 2-dimensional configuration (a 7*7 matrix) and the clones are numbered according to their position in this first configuration. Two additional configurations are obtained using the transforming matrices

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \quad A^2 = \begin{bmatrix} 5 & 3 \\ 3 & 2 \end{bmatrix}$$

For example the clone $c_{2,3}$ is put in position

$$\begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 0 \\ 5 \end{bmatrix} \\
 \begin{bmatrix} 5 & 3 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 5 \\ 5 \end{bmatrix}$$

in the second configuration and

in the third one. Clones are pooled according to the rows and the columns of each configuration (therefore there are 14 pools per configuration). Five positive clones (indicated in **bold**) are to be retrieved among the library. In each configuration, the clones that are selected as positive after comparison between the candidates from the current configuration and those from the previous ones are indicated with a plus (+), while the unselected candidates are indicated with a minus (-). In the first configuration there are 25 candidates and 25 selected clones, therefore 20 false positives. In the second configuration, there are 12 candidates and it remains 8 selected clones (5 true and 3 false positives) after analysis (that is after comparison between the candidates from the second configuration and those from the first one). The third configuration gives 12 candidates and allows retrieval of the 5 positives clones and elimination of the false positives. Forty-two tests have been needed to retrieve the positive clones. In that case the work saving is not very interesting, because there are only a few clones to analyse and a lot of positives with respect to the total number of clones. The section 4 and figure 3 show that the pooling strategy is much more advantageous when several thousands of clones are screened for retrieval of a few positive clones.

hyperplans in any scheme; if there is a scheme in which these two numbers are equal, then there is no false positive; 3) if these two values are not equal, there may be some false positives, but it is possible to detect them. Indeed, in any configuration they are inevitably at the intersection of hyperplans containing (at least) one other positive clone: if a clone is the only selected positive in a given hyperplan of a given configuration, then it is a true positive; otherwise it may be a false positive but the ambiguity is apparent; 4) it is always possible after analysis to test the dubious candidates in order to eliminate persistent false positives.

Another approach to the pooling strategy

It may appear difficult to imagine a configuration in a space of more than 3 dimensions, e.g. a 4-dimensional space. An equivalent, and perhaps more comprehensible approach would be to present the pooling strategy by assuming that clones are numbered with numbers expressed in basis $N^{1/D}$ (from 0000 to

aaaa where a equals $N^{1/4}-1$ in a 4-dimensional space), or are indexed with D coordinates, each varying from 0 to $N^{1/D}-1$. A pool is then built per coordinate: clones whose first coordinate is equal to 0 form a pool, those whose first coordinate is equal to 1 form another pool, ... and clones whose Dth coordinate is equal to $N^{1/D}-1$ form the last pool.

THE POOLING STRATEGY: PRACTICAL ASPECTS

Choice of the pooling dimension

Before the pooling strategy can be applied, the first problem is to choose the pooling dimension. This choice depends not only on the number of clones to be fingerprinted, the redundancy and the desired rate of false positives (fig. 1), but also on the technical possibilities: if the test to be performed is a PCR, it can be done on a maximum of several thousand YACs (3), and the pools have this size limitation; if the test to be performed is an hybridization

of a blot on which a digest has been deposited, the limit is about 150 YACs. In practice, since the number of clones per pool depends directly on the dimension (it equals $N^{(D-1)/D}$), those dimensions greater than 5 are usually not usable with STSs, and hybridizations restrict the dimension to 2 or 3. The best choice is the dimension given in figure 1, or if this is not technically possible, then the greatest possible dimension should be selected. In order to obtain smaller pools and use a larger dimension, half of the library can be arranged according to a configuration, and the other half according to another configuration (i.e. the pooling strategy can be applied independently on different sublibraries). Calculations have shown that this division would imply the use of more pools to obtain the same result in terms of the rate of false positives; thus this strategy should be avoided, unless technically indispensable (i.e. if the number of clones per pool is too large for the sensitivity of the detection method).

Reconfiguration problem: the transforming matrix

Once the dimension has been chosen, the next step is to determine the way in which pools will be assembled (i.e. the way the matrices, cubes or hypercubes are to be configured). The first configuration can be built in any way desired, the important point being the reconfiguration of the following configurations. In this paper, we examine a particular class of reconfigurations: the N clones from the library are arranged in an initial D -dimensional configuration, and we call $X = (x_i)$ the vector of their D coordinates on the matrix, cube or hypercube (we restrict the study to the square matrices and the regular cubes and hypercubes). We propose to rearrange the configurations by transforming the X coordinates of each clone in $X' = AX \% L$, where $A = (a_{ij})$ is a square matrix of integers, with size D , called the transforming matrix. ' $\% L$ ' means that the equality is verified modulo the configuration side length (the coefficients of X' equal the remainders of the division of the coefficients of AX by L). The reconfigurations of this class have the advantage of being entirely described by the transforming matrix and it is easy to derive some rules for selection of the transforming matrix which guarantee a satisfactory reconfiguration:

To be efficient in the elimination of false positives, a transforming matrix must have a determinant and all the subdeterminants non-null and prime with the configuration side length.

For example, the reconfiguration used by De Jong (5) is that obtained with the transforming matrix:

$$\begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$$

The determinant and all coefficients are non-null and prime with the side length if it is uneven, so the transformation is efficient for uneven side lengths. In 3 dimensions, an efficient transforming matrix would be for example:

$$\begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

if the side length is prime with 2 and 3.

A transforming matrix allows the second configuration to be built from the first one. If additional configurations are needed, some other transforming matrices must be chosen in order to obtain new configurations as different as possible from all the previous ones: if A_1 is the first transforming matrix, A_2 the

second one (which transforms the second configuration to the third configuration), ..., A_t the last one, the t transforming matrices are efficient in the elimination of false positives if, and only if, the selection criteria presented above is verified by the matrix $A_1 X A_2 A_1 X A_3 A_2 A_1 X \dots X A_t A_{t-1} \dots A_3 A_2 A_1$, where $A X B$ designates the matrix:

$$\begin{bmatrix} A \\ B \end{bmatrix}$$

Of course, it is possible to always use the same transforming matrix ($A_1 = A_2 = A_3 \dots = A$): in this case, if C configurations are used, the rule of selection of the transforming matrices must be verified by $A \S A^2 \S \dots \S A^{C-1}$. For example, if three 2-dimensional configurations are needed, the choice

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$$

is good provided that the side length is prime with 2, 3 and 5, since all subdeterminants of $A \S A^2$ are equal to 1, 2, 3 or 5:

$$A \S A^2 = \begin{bmatrix} 2 & 1 \\ 1 & 1 \\ 5 & 3 \\ 3 & 2 \end{bmatrix}$$

The figure 2 gives an example of reconfiguration using three configurations obtained from these two transforming matrices.

Choice of the side length

Theoretically, the side length can be taken to be equal to $N^{1/D}$ (or to the first integer greater than $N^{1/D}$ if $N^{1/D}$ is not an integer). However this choice can create problems: e.g., if the side length is even, all coefficients of the transforming matrix must be uneven, but in this case the determinant would be even and the rules of selection cannot be verified. It clearly appears to be of interest to choose the side length from among the prime numbers, in which case, it will be more easily prime with all subdeterminants of any transforming matrix. However, this would mean that the side length would generally be larger than $N^{1/D}$ (which is in any case true when $N^{1/D}$ is not an integer) and the configurations will thus contain more than the required number of clones N . This implies that some part of each configuration would remain empty. To maximize the efficiency, these blanks have to be distributed as uniformly as possible in every configuration. An easy way to achieve this would be to use a $C+1^{\text{th}}$ virtual configuration built with a C^{th} efficient transforming matrix and to arrange all the blanks in a minimum number of hyperplans. The location of the blanks in the previous configurations can then be deduced and this method guarantees that they will be uniformly distributed in every pool of the effectively used configurations.

Choice of the number of configurations

This choice depends on the desired rate of false positives. It is possible use only a small number of configurations and to verify a large number of selected positive clones individually. However, it may be preferable to use more configurations (and therefore to initially assemble more pools and perform a larger number of tests) in order to reduce the tedious task of individual clone verifications. In practice, the choice of the number of configurations is a compromise between these two considerations. Note that it is possible to use only some of the schemes from

	Number of Clones : 72,000						Redundancy : 10			
	Multi-Configurations D-Dimensional Pooling Strategy									Classical method
pooling dimension	D = 2			D = 3			D = 4			94 sublibraries of 8 microplates + 28 pools per sublibrary
configuration side length	L = 269			L = 43			L = 17			
number of configurations	c = 1	c = 2	c = 3	c = 2	c = 3	c = 4	c = 4	c = 6	c = 8	
number of pools	538	1076	1614	258	387	516	272	408	544	2726
number of tests	538	1076	1614	258	387	516	272	408	544	374
number of false positives	95	0,13	0,0003	14,3	0,31	0,008	5,6	0,17	0,009	3,3
number of additional tests	105	10	10	24	10	10	16	10	10	4,3
number of library copies	2	4	6	6	9	12	16	24	32	4
number of clones per pool	269			1849			4913			768, 96 and 64

Fig. 3. Comparison between pooling strategies using different dimensions and different numbers of configurations, and a classical method (see text) for the screening of a 10-fold representative library containing 72,000 clones. Results are given in terms of the total number of pools to be assembled, the number of tests to be performed for one screening, the expected number of false positives, the number of additional tests required to eliminate the false positives, the number of copies of the library needed and the number of clones per pool. Pooling strategies require the assembly of a much smaller number pools than would a classical method. The number of tests depends on the acceptable rate of false positives: if one is prepared to verify the positive clones in order to eliminate the false positives, then two 3-dimensional configurations will be sufficient to screen the library; in order to avoid this tedious task, four 3-dimensional configurations would be a better choice. Note that the results when using 4-dimensional configurations are also interesting. However this method requires that PCR be performed on pools of 4913 clones, which could prove problematic. Besides, it would appear more reasonable to use a 3-dimensional approach because if the redundancy of the library were underestimated, the results of the 4-dimensional strategy would very quickly diverge.

a configuration. For example, if the optimal dimension of a library is 3, but the desired rate of false positives is intermediate between the rate obtained with two and three 3-dimensional configurations, two 3-dimensional configurations can be used, plus one or two schemes from a third 3-dimensional configuration.

Automation

The pools corresponding to the first configuration can be assembled manually, as the first configuration can be chosen in such a way as to simplify pooling (pools may correspond to microplate rows or columns, or to entire microplates). However it is obvious that for the following reconfigurations, manual clone pooling will be a puzzle that cannot be envisaged.

At this point appears the main drawback of the pooling strategy, i.e. that the setting up of a large number of clones, even in two-dimensional pools, cannot be considered without automation. Therefore the development of a well-suited robot should be very useful (7).

RESULTS AND PERSPECTIVES

Example of library screening

It may be useful to give an example to illustrate these theoretical considerations: we could envisage using the pooling strategy to screen the CEPH YACs library (at present 72,000 clones, redundancy 10, mean insert length 500 kb). Figure 1 indicates the optimal pooling dimension as 3; thus the side length should

be equal to $72,000^{1/3} = 42$ but as this is an even number it is preferable to use 43 which is uneven and a prime number. The transforming matrix

$$A = \begin{bmatrix} 7 & 4 & 1 \\ 3 & 7 & 4 \\ 2 & 3 & 7 \end{bmatrix}$$

is efficient for this side length, as are A , A^2 and A , A^2 , A^3 but not A , A^2 , A^3 , A^4 . Therefore 4 configurations can be built efficiently with A . Figure 3 presents the expected results in terms of the number of configurations used, the number of pools used, the number of tests to be performed, the number of clones per pool and the expected number of false positives for dimensions 2, 3 and 4.

In 3 dimensions, only 2 configurations and 258 pools (and therefore 258 tests for one screen) are needed in order to obtain a mean of 10 true positives and 14,3 false positives. The 2-dimensional strategy is clearly less efficient, while the 4-dimensional strategy is almost equivalent (272 pools, 5.6 false positives) but the pool size is larger and could create problems (4913 clones per pool with 4 dimensions while only 1849 with the 3-dimensional strategy (a reasonable number for a PCR)).

Finally, we can compare the pooling strategy to a modification of a previous screening method. The library to be screened can be divided into several sublibraries [the CEPH library (750 microplates) could be divided in 94 subgroups of 8 microplates], the positive sublibraries are first determined and then a 3-dimensional approach with only one configuration for example can be used for each sublibrary in order to locate the positive

clone. This method requires that 2726 different pools be assembled and an average of 374 tests be performed. The number of tests is slightly higher and the number of pools is dramatically increased with regard to the multi-configurations pooling strategy. Besides the classical strategy always comprises two steps and it is difficult to systemize the screening since the second step (the screening within the positive sublibraries) depends on the first one (the screening of all the sublibraries). On the other hand the multi-configurations pooling strategy allow easily a single hit screening with just a few verifications.

Application to physical mapping

It has already been proposed to fingerprint random clones with single copy landmarks extracted randomly from a region of interest in order to construct a physical map (8, 9). For example the use of the CEPH YAC library fingerprinted with 1750 STSs extracted from a 150 Mb chromosome would lead to 20 contigs with an average size 6.7 Mb. The pooling strategy can be helpful for this method, since it allows optimization of library screening. Indeed while a direct approach would require 126 millions of tests, the 3-dimensional pooling approach reduces the number of tests to about 903,000 (when using 4 configurations). Obviously, false positives can occur in the resultant map but these can be detected when they create inconsistency in the map (in the example presented above, only one false positive should persist in the final map after elimination on map inconsistency criteria only) and secondly they can be eliminated after verification of the presumed positive clones (a few thousand additional tests would be needed).

Physical mapping with a double-pooling strategy

Objections can be raised that the 903,000 tests of a 150 Mb chromosome mapping project represent a considerable amount of work, but again it is possible to minimize this work. The pooling strategy can also be applied to STSs: STSs should be arranged in a matrix and pooled according to the rows and the columns. Each STS pool is then tested on each clone pool. Similar to clone pooling, several matrices may be needed in order to eliminate illegitimate STS candidates. Analysis of the test results of all the STS pools on a given clone pool permits to ascertain for which STS the clone pool is positive, and the previously described problem of single clone pooling again recurs.

However, this double pooling technique has three principal limitations: first, it is difficult to achieve in practical terms, since it is not possible to pool a large number of STSs in the same group without encountering problems with oligomer crosshybridization or PCR efficiency (thus the 2-dimensional approach will be the only one usable). Secondly, when there is a large number of clones per pool, each STS pool has a high probability of being positive for a given clone pool. Thus, the number of positive STSs per STS matrix and therefore the number of STS matrices required to eliminate the illegitimate STS candidates will be high, leading to a high number of STS pools. Thirdly, calculations show that when a double pooling strategy is used, it is often more advantageous to use a single pooling strategy with a higher dimension (unless a higher dimension is technically impossible to use). These three remarks imply that a double pooling strategy is not always interesting but that the economy in terms of work (up to 60%) can be considered in some cases.

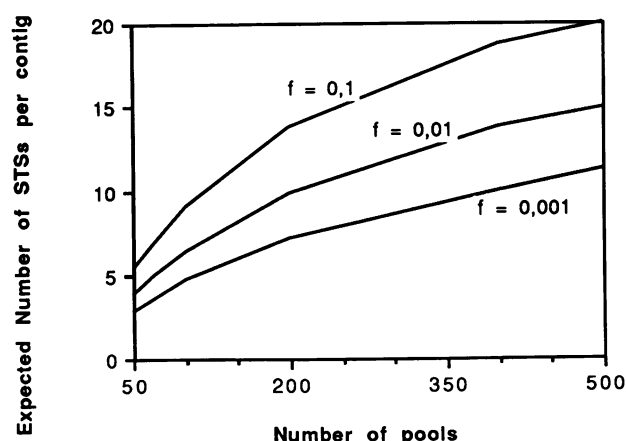


Fig. 4. Expected number of STSs per contig as a function of the number of pools for different rate of false positives. The curves are given for the mapping of 50 STSs randomly extracted from a 5 Mb region and tested on the CEPH YAC library (500 kb inserts, redundancy 10) divided into 50 to 500 pools. For example if the library is divided into 350 pools, 3 contigs of 17 STSs are expected if a probability of 0.1 of having a false positive is tolerated.

Single scheme pooling strategy applied to STS mapping

Instead of multiplying the configurations in order to eliminate false positives, it is possible to use only one scheme (that is to use only one copy from the library and to divide it into pools or sublibraries) and to test each STS on every pool from the scheme. For each STS, we obtain a fingerprint that consists in a list of positive pools. The fingerprints of each STS pair are compared and a likelihood ratio is calculated (the ratio of the probability that the two STSs have at least one common clone knowing the data to the probability that the two STSs have no common clone knowing the data): if this likelihood ratio is above a given threshold, the two STSs are put in the same contig. Therefore we obtain some STS contigs, whose size depends on the tolerated rate of false positives (a false positive occur when two STSs having no common clone have a likelihood greater than the threshold), the number of pools from the scheme, the number of STSs, the redundancy of the clone library and the size of the region to be mapped. The figure 4 shows the average STS contig size as a function of the number of pools in the scheme for the 500 kb clone library from CEPH, fingerprinted with 50 STSs extracted randomly from a 5 Mb region. For example, by screening 350 pools with 50 STSs on this library, 17,500 tests would be needed to obtain 3 contigs of about 17 STSs. If larger insert libraries would be obtained (for example 1 Mb YACs) with the same redundancy, by screening 115 pools with 25 STSs, 2,875 tests would be necessary to obtain 2 contigs of 12 STSs. In both cases, the expected number of false positives would be 0.1.

Obviously this single scheme pooling strategy has some drawbacks: 1) all information on individual clones is lost and we only build STS contigs 2) the contigs are smaller than in the case where the screening allows retrieval of the positive clones. On the other hand there are some advantages: 1) only one copy of the library is needed 2) pools are very easy to assemble because they can correspond to microplates or groups of microplates. Therefore this method does not require automation of the pooling assembly 3) this method allows a rapid STSs ordering: the contigs obtained are small but they require only a low number of tests:

therefore this method, when used with very large YACs, should be very useful for genetic mapping because it should allow to order polymorphic STSs without any knowledge about the genetic distances. A small number of meioses would then be sufficient to estimate roughly these genetic distances. For a given number of STSs, the precision on the STS order is depending on the redundancy and the insert size. If large YACs (> 1 Mb) can be obtained, the combined screening of DNA families and YAC DNA pools would allow an integrated construction of both genetic and physical maps of the human genome.

MATHEMATICAL PROOFS

Only two problems should require explanation:

Computation of the expected number f of false positives

Two approaches have been considered: 1) we derived the expected number of false positives assuming random reconfigurations, which gives a pessimistic, but good approximation, 2) we used the Monte-Carlo simulations in order to evaluate the expected number of false positives among N clones with C configurations in D dimensions. For methods 1) and 2) we assumed the number of positive clones per screening to have a Poisson distribution. Our programs allow computation for any value of C , D and N and verifies the efficiency of the transforming matrices. The two methods give consistent results: for four 3-dimensional configurations with side length 43 and a redundancy 10, the expected numbers of false positives are respectively 0.008 and 0.012. The numerical results from the figure 3 were obtained with the Monte-Carlo method, while we assumed random reconfigurations for figure 1.

Rules of selection of the transforming matrix

First each transforming matrix must perform a one-to-one mapping from the incoming configuration to the outgoing. As the transformation $X \rightarrow AX \% L$ is linear, this condition is equivalent to $(AX \% L = 0 \Rightarrow X = 0)$. Thus it is easy to derive that this expression is equivalent to the expression: 'the determinant of each transforming matrix must be non-null and prime with the configuration side length'.

Secondly the transforming matrices must modify the configuration as much as possible: they must be efficient. Two elements with common coordinates in a given configuration must have no common coordinates in another configuration. In fact this requirement is generally impossible to achieve: for example, all clones from a given hyperplan of a first configuration have one common coordinate and must therefore be put in different hyperplans in a second configuration; but with the exception that in two dimensions, there are not enough different hyperplans to ensure that all the clones are in different hyperplans in a second configuration. It is only possible to guarantee that two clones will have at the most $D-1$ common coordinates in all the configurations (if the number of configurations is not too large). We will term 'efficient' any group of transforming matrices that gives outgoing configurations in which this condition is verified. Using the linearity of the transformation, it is straightforward to derive the conditions of efficiency:

A_1, A_2, \dots, A_t are efficient for a side length L if, and only if, all subdeterminants of $A_1 A_2 A_1 A_3 A_2 A_1 \dots A_t A_{t-1} \dots A_3 A_2 A_1$ are non-null and prime with L .

Note that 1) matrices can be efficient for one particular side length and not for another; 2) the conditions of efficiency imply the conditions of one-to-one mapping.

ACKNOWLEDGEMENTS

This work was supported by the Ministère de la Recherche et de la Technologie and the Association Française contre les Myopathies.

REFERENCES

1. Albertsen, H.M., Abderrahim, H., Cann, H.M., Dausset, J., Le Paslier, D. & Cohen, D. (1990) Proc. Natl. Acad. Sci. USA 87, 4256–4260
2. Olson, M.V., Hood, L., Cantor, C. & Botstein, D. (1989) Science 245, 1334–1335
3. Green, E.D. & Olson, M.V. (1990) Proc. Natl. Acad. Sci. USA 87, 1213–1217
4. Kwiatkowski, T.J. Jr, Zoghbi, H.Y., Ledbetter, S.A., Ellison, K.A., Craig Chinault, A. (1990) Nucleic Acids Res. 18, 7191–7192
5. De Jong, P.J., Aslanidis, C., Alleman, J. & Chen, C., (1990) Genome Mapping and Sequencing, Cold Spring Harbor, New York, p48
6. Evans, G.A. & Lewis, K.A. (1989) Proc. Natl. Acad. Sci. USA 86, 5030–5034
7. Cohen, D., (1991) Genome Mapping and Sequencing, Cold Spring Harbor, New York, p247
8. Barillot, E., Dausset, J. & Cohen, D. (1991) Proc. Natl. Acad. Sci. USA 88, 3917–3921
9. Torney, D.C. (1991) J. Mol. Biol. 217, 259–264